2009年度修士論文

検索エンジンのヒット数に対する信頼性の検証

早稲田大学大学院基幹理工学研究科

情報理工学専攻

舟橋　卓也

学籍番号　5108B112-6

提出　2010年 2月 1日

指導　山名 早人　教授

概要

近年，Web 検索エンジンを本来の用途である検索以外に，翻訳支援を始めとした自然言語処理などのアプリケーションへの活用を試みる研究が行われている．これらの研究の多くは，クエリに対する検索結果のヒット数を利用している．ここで，ヒット数とは検索エンジンが返す，クエリに合致する Web ページ数を指す．これらの研究では，検索エンジンが返すヒット数は信頼できるという仮定の下で用いられてきた．しかし，実際にはヒット数は検索するタイミングによって値が変化することが知られており，ヒット数の信頼性に対しては疑問が残る．もしも検索エンジンのヒット数が信頼できないものであるならば，ヒット数を利用した研究結果の信頼性にも同様に疑問が残る．そのため，検索エンジンのヒット数の特徴を明らかにし，その信頼性を示すことは重要である．これまでに，ヒット数の変動傾向を明らかにするためにいくつかの研究が行われてきた．しかし，どのような状況下ならヒット数が信頼できるのかについて論じた研究はこれまでに存在しない．そこで本研究では，検索エンジンのヒット数が変化する 3 つのケース，（1）短時間に繰り返し検索を行った場合，（2）検索開始オフセットを変化させた場合，（3）検索を行う日時が変わった場合，について，それぞれヒット数の変化の特徴・信頼性を示す．その上で，どのような状況下で得られたヒット数が信頼できるのかについて述べる．その結果として，信頼できるヒット数は「検索開始オフセットが最も大きい場合に得られるヒット数で，かつ，検索エンジンによってヒット数の調整が行われていない場合」かつ「時間経過による変化が 1 週間以上安定している場合」に得られることを確認した．

ABSTRACT

In this paper, we provide a scientific basis to adopt search engines' hit counts, numbers returned as search result counts. Since many studies adopt search engines' hit counts to estimate the popularity of a particular query, the reliability of hit counts is indispensable for archiving trustworthy studies. However, hit counts are unreliable because they will "dance," i.e., change, when a user clicks the "Search" button more than once or clicks the "Next" button on the search results page, or when a user queries the same term on a another day. In order to provide a scientific basis to adopt search engines' hit counts, we have analyzed the characteristics of hit count transition by gathering various types of hit counts over two months by using 10,000 queries. On the basis of our experiment, we have concluded that the last hit counts, i.e., hit counts with the largest search offset just before search engines adjust, are more reliable than the hit counts obtained with the top 10 results. Moreover, hit counts are reliable when they are consistent over a period of approximately a week.

目次

1. はじめに

近年，手軽に Web から情報を取得するために，Web 検索エンジンが広く利用されるようになった．Web 検索エンジンは，検索フォームよりクエリキーワードを受け取り，そのクエリキーワードを含む Web ページ集合とその数を検索結果として返すシステムである．従来，Web から情報を取得するためには，各自で Web をクローリングするか，リンクを辿り求める情報を探すしか方法がなかった．しかし，検索エンジンの登場によって，ユーザは検索エンジンにクエリキーワードを入れるだけで Web 上の情報を獲得することができるようになった．検索エンジンの登場によって，手軽に Web 上から情報を収集することができるようになったため，検索エンジンの検索結果を利用して研究を行う試みが行われている．そのなかで，検索エンジンの結果数（以後，ヒット数と呼ぶ）を利用した研究が広く行われている[1][2][3]．検索エンジンのヒット数は，検索エンジンが収集した全 Web ページにおいてクエリキーワードが出現する Web ページ数の概算とみなすことができる．そのため，ヒット数を利用して，機械翻訳の精度向上を試みる研究[1]や，クエリ単語間の距離を定義する研究[2]，単語クラスタリングを試みる研究[3]などが存在している．

このように広く用いられている Web ヒット数であるが，その信頼性は明らかになっていない．例えば，検索エンジンは様々な状況下で変化をする．具体的には，ヒット数が変動する例として次のような３つのケース (1) 「検索」ボタンを何度もクリックした場合，(2) 検索開始オフセット[1]を変更し，検索を行った場合，(3) 検索する日時を変えて検索を行った場合，が挙げられる．このようにヒット数は様々な条件下で変動をするため，どのような状況におけるヒット数が信頼できるのか明らかとなっていない．

そのため，これまでにヒット数の変動傾向を明らかにするため，いくつかの研究が行われてきた[4][5][6]．しかしながら，これらの研究ではヒット数の信頼性について論じておらず，どのようなヒット数が信頼できるのか明らかにされていない．ヒット数の信頼性が明らかでなければ，ヒット数を利用した研究についても信頼性にも疑問が残ることとなる．

そこで本研究では，ヒット数の変動の特徴について検証した後，どのような状況下で得られるヒット数が信頼できるのかについて論じる．具体的には，ヒット数が変動する３つのケースそれぞれについて，実際に 10,000 個のクエリキーワードを利用し Hit Count Dance の特徴について検証を行う．検証は，Google，Yahoo!，Bing がそれぞれ提供している検索 API を通して行う．

以下，本論文では次の構成をとる．まず，２章において関連研究について述べる．次に，３章において，ヒット数の検証方法について述べ，続いて４章において検証結果について述べる．５章において，検証の結果から信頼できるヒット数について論じ，最後に６章においてまとめを述べる．

[1] 検索開始オフセットとは，検索をする際に指定することのできる，結果として表れる Web ページの内でもっとも高いランキングの順位を指す．

2. 関連研究

本章では，関連研究として以下の３項目について述べる．

・検索エンジンの構成
・検索エンジンのヒット数を利用した研究
・ヒット数の変動傾向を調査した研究

本論文では検索エンジンのヒット数に対して検証を行うことを目的としている．そのため，前提知識として現在商用検索エンジンで一般的に利用されていると考えられる技術について 2.1 で述べる．続いて，ヒット数に対する信頼性の検証を行う意義を示すために，これまでにどのような研究においてヒット数が利用されてきたのかについて 2.2 で述べる．また，本研究の類似研究として検索エンジンのヒット数がどのように変動をするのか調査をした研究について 2.3 で述べる．

2.1 検索エンジンの構成

本項では検索エンジン，中でも現在商用検索エンジンで主に利用されている全文型検索エンジンの構成について述べる．まず，2.1.1 で全文型検索エンジンの代表的な構成について述べたのち，2.1.2 にて現在の検索エンジンで利用されている検索の高速化技術について述べる．

2.1.1 全文型検索エンジンの代表的な構成

Arasu ら[8]によって執筆された検索エンジンに関する調査論文によれば，検索エンジンは大きく分けて「Crawler」と「Indexer」に分けることができる．初めに，Arasu らの論文を参考に全文型検索エンジンの構成についてまとめた図を図 1 に示す．

「Crawler」とは，検索エンジンが索引を作成するための元となる Web ページを収集するプログラムである．Web は管理された空間ではないため，世の中には Web ページの一覧という情報は存在していない．そこで検索エンジンは Web ページの索引を作成するために，独自に Web ページを収集する Crawler と呼ばれるプログラムを利用している．Crawler は，シードとして与えられた Web ページからハイパーリンクを辿ることによって，Web ページを収集するプログラムである．全文検索型検索エンジンにおける Crawler の目的は全世界の Web ページの収集である．しかし，400 億ページ以上存在しているといわれる Web をすべて収集することは，時間的な制約や計算資源の制約によって実現することが非常に難しい．そこで，Crawler は「ユーザが求めるページ」を効率よく収集するアルゴリズムに従って動作している．例えば，検索エンジンのユーザは PageRank[4]の高いページを検索結果として

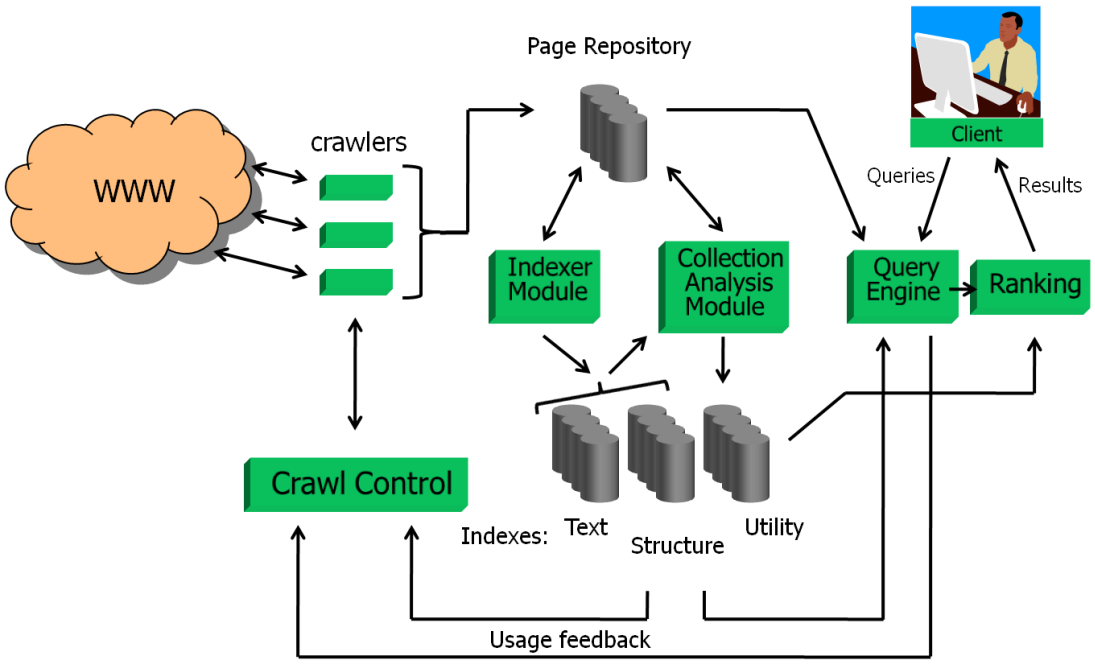


図 1. 全文型検索エンジンの全体構成（[8]より引用）

求めることが知られており，ハイパーリンクを辿る際にリンク先 Web ページの PageRank を見積もりながら．PageRank の高いであろう Web ページを優先的にクロールするアルゴリズムが考案されている．

「Indexer」は Crawler が収集した Web ページを解析し，クエリとして入力されるキーワードで検索ができるよう索引を構築するプログラムである．Indexer は転置インデックスと呼ばれる，キーワードと文書を関連付けた索引を作成し，高速な検索を実現している．

2.1.2 検索エンジンをより高速化するためのアルゴリズム

検索エンジンは，Crawler が収集したページに対して転置インデックスを作成することで高速な検索を実現している．しかしながら，検索対象となる Web 文章が膨大であること，検索エンジンを利用するユーザが増加したことなどが原因となり，単純な構成ではユーザーエクスペリエンスの高い検索エンジンを提供することができなくなった．そこで，検索エンジンでは分散処理[9]，early termination[10]，index pruning[12]をはじめとする技術を利用することで，検索の更なる高速化を図っている．

分散処理とは，その名の通り検索のプロセスを複数台のマシンで実行することである．転置インデックスを複数のファイルに分割した後，分割したファイルを複数台のマシンに設置をすることで，マシン 1 台あたりの処理量を分散させることができる．

early termination とは，検索結果の質を落とさずに「手を抜く」ことによって，高速に検索を実行するアルゴリズムである．Jansen らによる研究[11]では，ほとんどの検索エンジンユーザは最大でもランキングの上位 20 件までしか閲覧しないと報告がされている．そこで，ランキング上位の結果を高速に取得するために，early termination では一定数の検索結果が集まった時点で検索プロセスを打ち切り，打ち切った時点で集まった結果から検索結果を提示する．early termination を利用することで，ランキングが上位の結果を高い精度で検索結果として提示しながら，検索にかかる計算コストを抑えることができる．

index pruning とは，転置インデックスの大きさを制限することで，高速に検索を行う技術である．先ほど述べたように，検索エンジンにおいては多くのユーザが検索結果のランキング上位の結果しか閲覧をしない．そのため，転置インデックスの中には静的ランキングの高いドキュメントが一定数あれば，ほとんどの場合検索を行うことができる．そこで，index pruning では，インデックスのサイズに制限をかけ，保持するデータ量の削減と高速な検索を実現している．

このように，検索エンジンはユーザの望む結果を返しながら，高速に検索を行うためのアルゴリズムを導入している．これらのアルゴリズムはほとんどすべてのケースにおいて，ユーザーエクスペリエンスの向上につながる．しかしながら完全な転置インデックスを放棄してしまうため，「ランキングが下位の全ての検索結果を取得したい」，「正確なヒット数を求めたい」といった要望には対応ができなくなる副作用も持つ．

2.2 検索エンジンのヒット数を利用した研究

本項では，検索エンジンのヒット数を利用した研究について例を挙げる．

2.2.1 Web コーパスとしての利用

検索エンジンのヒット数は，検索エンジンが収集した全 Web ページにおいてクエリキーワードが出現する Web ページ数の概算とみなすことができる．それは同時に，検索エンジンのヒット数は，検索クエリキーワードの Web に対する Document Frequency と見なせることを意味する．Kilgarriff[1]らは，Web 上には非常に多くの文章が存在しており，検索エンジンを利用することで Web をコーパスとして利用することが可能だと述べている．

2.2.2 Google Similarity Distance

Cilibrasi[2]らは検索エンジンのヒット数を利用した単語間の類似度 Google Similarity Distance を提案した．検索エンジンにおいて AND 検索を利用することで，単語間の共起度を取得し単語の類似度を算出している．Google Similarity Distance も Web をコーパスとして利用することで類似度を算出している．

2.2.3 ワードクラスタリングへの利用

松尾ら[3]は，Web をコーパスとみなし，ヒット数からキーワード間の類似度を算出することでワードクラスタリングを行う手法を提案している．松尾らは２つのキーワードを，NOT 演算子を利用しながら検索をすることで，単語間の距離を定義した．定義した距離を利用してクラスタリングを行った結果，概略としてのクラスタを作成するには十分な精度を得ることができた．松尾らの手法ではヒット数を利用しているため新語にも対応がしやすいというメリットが存在する．

2.3 ヒット数の変動傾向を調査した研究

2.2 で述べたように，これまでにヒット数は広く用いられてきた．しかしながら，これらの研究で利用されてきたヒット数は変動することが知られており，その変動傾向を明らかにするためにいくつか研究が行われてきた．本節では，ヒット数の変動について研究を行った論文について，その概要を述べる．

2.3.1 Kilgarriff による研究

Kilgarriff[4]は，Web をコーパスとして利用した自然言語処理[1]を行ってきた．その過程で，検索エンジンのヒット数が変化をすることに対して疑問を持ち，ヒット数の変動傾向について調査を行った．特定のクエリを利用して 1 ヶ月間毎日ヒット数の取得を行った結果，前日と比較して 10%以上変化をした日数が 9 日 100%以上変化をした日数が 6 日存在した．この結果を踏まえて Kilgarriff は，検索エンジンを利用して研究を行うためには，検索エンジンの信頼性について明らかにする必要があると提言を行った．

Kilgarriff は検証実験について，自身の論文の中で「小規模の実験を行った」とだけ述べており，どのようなクエリを利用し，どのような環境下で実験を行ったのかについて言及をしていない．しかしながら，自然言語処理の研究者である Kilgarriff が，ヒット数が変化をすることについて報告をしその信頼性を明らかにする必要性があることの提言を行っていることは，自然言語処理分野において検索エンジンの有用性が高まっているためだと考えられる．

2.3.2 Thelwall による研究

Thelwall[5]は，Google，Yahoo!，Live Search から得られるヒット数をそれぞれ比較することでヒット数の正確さについて検証を行った．実験は 1 つの単語からなる 1,587 個のクエリを使用し，検索エンジン各社が提供している API を利用して検索を行った．

それぞれの検索エンジンにおいてにヒット数を取得をしたところ，ヒット数の大きさ順は検索エンジン毎におおむね一貫性を保っていた．しかしながら，ヒット数の大きさの絶対値は Google と Yahoo!において Bing の 5 倍から 6 倍のヒット数が得られた．Yahoo!は Google と比較するとわずかにヒット数が多くなったものの，その原因は「Yahoo!が Google よりも拡張範囲の広いクエリ拡張を行っているため」と Thelwall は考察している．

また，それぞれの検索結果についても分析を行ったところ，Yahoo!は Google に比べてわずかに多様な検索結果を返すことが分かった．Live Search では他の２つの検索エンジンと比較すると，多様性の低い検索結果しか得ることができなかった．ここで，検索結果の多様性は「検索結果に含まれる URL」，「検索結果に含まれるホスト数」，「検索結果に含まれるトップレベルドメイン数」を利用して判断を行っている．

この結果より，Thelwall らはヒット数を「検索クエリキーワードの Web に対する Document Frequency」と見なして利用する研究には Google を，検索結果をコーパスとして利用する研究には Yahoo!が適していると述べている．

このように，Thelwall は検索エンジンの特徴を調査し，どの検索エンジンが，どのような研究に対して適しているかについて研究を行った．しかしながら，Thelwall らの研究では時間経過によるヒット数の変動・検索結果の変動について触れられておらず，十分な調査であるとは言うことができない．

2.3.3 Uyar による研究

Uyar[6]は，Thelwall と同様に Google，Yahoo!，Live Search から得られるヒット数の変動に対して調査を行った．Uyar は「実際に得られる検索結果数とヒット数との差異」と「時系列によるヒット数の変化」という２つの視点からヒット数の変動傾向を調査した．

実際に得られる検索結果数とヒット数の差異を利用した検証

現在，一般的な商用検索エンジンは，検索結果として最大でランキングの上位 1,000 件まで Web ページを返す．Uyar は，実際に取得することのできた Web ページ数が 1000 件以下の場合, 実際に得られた Web ページ数が正確なヒット数であると仮定を行った. その上で, 得られたヒット数と正確なヒット数の間の誤差を定義した．正確なヒット数を *ReturnedDocuments*, ヒット数を *Estimate* とした時の誤差 *Error* の定義を数式 2.3.3.1 に示す.

$$Error = Estimate - ReturnedDocument \quad \text{... (数式 2.3.3.1)}$$

加えて Uyar らは, Error を元にエラー率 Percentage of Error を数式 2.3.3.2 のように定義した.

$$Percentage\ of\ Error = 100 \times \frac{Error}{ReturnedDocument} \quad \text{... (数式 2.3.3.2)}$$

このエラー率を利用し, Uyar は Hit 数の誤差について検証を行った結果, Google では 78%のクエリにおいてエラー率が 10%以下，Yahoo!では 43%のクエリにおいてエラー率が 10%以下となった．しかしながら，Yahoo!においては 53%のクエリにおいてエラー率が 50%以上の値になっており，Google に比べると Yahoo!のエラー率は高いことが明らかとなった．

このように Uyar はヒット数のエラー率の定義を行ったものの，その前提として実際に得られた Web ページ数が正確なヒット数であると仮定を行っている．しかしながら，実際に得られた Web ページ数は必ずしも正確なヒット数とは述べることができない．なぜなら，検索エンジンは高速な検索を実現するため 2.1 で述べた技術を利用しており，高速化の副作用として検索結果として表示する Web ページ数に制限ができていると考えられるためである．

時系列によるヒット数変動に対する調査

Uyar は検索結果として表れた Web ページを利用してヒット数の特徴分析を行うとともに，時間経過によってヒット数がどの程度変化をするのかについても調査を行っている．調査を行った結果，平均的には Google の変化量が最も少ないこと，Yahoo!ではヒット数の値が大きくなるほど時系列による変化量が小さくなること，Live Search では毎日 10%以上の変動が発生していることなどが明らかになった．Uyar の時間経過によるヒット数の変化は，日々の変化の割合は出しているものの，どのような場合ならば信頼できるヒット数なのかについては言及しておらず，ヒット数の信頼性は明らかにしていない．

2.3.4 ヒット数の変動傾向を調査した研究のまとめ

ヒット数の変動傾向調査に関する研究についてまとめた表を，表 2.1 に示す．

表 2.1 ヒット数の変動傾向を調査した研究

	着眼点	問題点
Kilgarriff	・ヒット数が変動を行うことに対する問題提起	実験が小規模である点
Thelwall	・検索エンジン間におけるヒット数の大きさの比較 ・検索エンジン間における検索結果の多様性の比較	時間経過によるヒット数変化を考慮していない点
Uyar ら	・実際に取得できた検索結果数とヒット数の比較	根拠となっている「実際に取得できた数」が信頼できるのかどうか疑問が残る点
	・時間経過によるヒット数の変動割合	どのように「時間経過によるヒット数の変動」を克服し，信頼できるヒット数を得るのか述べられていない点

3. 検証方法

3.1 検証項目

本研究では，検索エンジンのヒット数に対する信頼性を検証するため，次に示す検索エンジンが変動をする 3 つのケースにおいて，どのようにヒット数の変動が発生するのか検証を行う．

ケース 1：短時間に，繰り返し同じクエリを利用して検索し場合
ケース 2：短時間に，繰り返し「次へ」ボタンをクリックした場合
ケース 3：検索を行う日時を変えた場合

本検証においては，Yahoo! Japan の 2007 年 12 月のクエリログにおいて頻出順に並べて現れた上位 10,000 件をクエリとして利用した．このクエリログは，情報爆発時代に向けた新しい IT 基盤技術の研究[13]において提供頂いたデータセットである．このデータセットにおけるクエリの単語数分布を表 3.1 に示す．ここで単語とは，スペースで区切られた連続文字列を差す．利用したクエリのうち，200 個のサンプルを付録 I に示す．

Google，Yahoo!，Bing が提供している検索 API に対して上記 10,000 個のクエリで検索することにより，ヒット数の変動に対する特徴を検証する．検索 API を利用するにあたって，各 API の設定はそれぞれの検索エンジンが設定をしているデフォルト設定を主に利用した．検索時に利用した各検索 API の設定を表 3.2 に示す．表 3.2 中の赤文字は，デフォルト設定を変更している項目を表す．ただし，ケース 2 における実験においては，一度に取得する検索結果数をそれぞれ 10 に設定をし実験を行った．

本検証では 10,000 個のクエリを利用して検索を行った．しかし，検索の過程においてエラーが発生し検索結果が取得できない場合があった．そこで，エラーが発生し，検索結果が取得できなかった場合，そのクエリによる検索結果を取り除いた上で検証を行った．

最後に本検証は，2009 年 10 月から 2009 年 12 月にかけて行った．

表 3.1 クエリの単語長分布

クエリ中の単語数	頻度
1	9,522
2	434
3	42
4	1
5	1
合計	10,000

表 3.2 検索時における各検索 API の設定

	Google	Yahoo!	Bing
一度の取得する検索結果数	8	50	50
検索元の言語	日本	指定なし	日本
セーフフィルタ	中	無し	中

3.2 短時間に，同じクエリを利用して検索した場合に発生する変動

何度も「検索」ボタンを押すことによって発生するヒット数の変動幅を調べるために，変動係数(*CV*)を利用する．*CV* は標準偏差を平均を利用して正規化したものであり，次の数式 3.2 で定義される．

$$CV = \frac{\text{標準偏差}}{\text{平均}} = \frac{\sqrt{\text{分散}}}{\text{平均}} \quad \text{数式 3.2}$$

CV は標準偏差を正規化した値であるので，*CV* を求めることにより，各検索エンジンにおいて平均に対して何%の変動が発生しているのかを求めることができる．

実験では，全ての検索エンジンに対して，10,000 個のクエリ各々について 100 回ずつ，5 分以内に検索を行い，取得した 100 個のヒット数から *CV* を算出する．短時間に何度も検索を行う理由は，検索エンジンのインデックス更新によるヒット数の変動から受ける影響を避けるためである．

3.3 検索開始オフセットの変化に伴う変動

短時間に「次へ」ボタンを繰り返しクリックしたことによるヒット数の変動を検証するために，10,000 個のクエリそれぞれについて *HitCount*(1,10)，*HitCount*(11,20)，..., *HitCount*(991,1000)を取得する．ここで，*HitCount*(*k*, *k*+9)は，一度に取得する検索結果数を 10 に設定し，*k* 番目にランキングされた結果から検索結果を取得した時のヒット数とする．なお, この k は検索開始オフセットと等しい値となる. *HitCount*(1,10)から *HitCount*(991, 1000)を取得した後，Hit Count Dance の傾向を捉えるため *Deep hit count Vector* (*DV*)を定義する．*DV* の定義を数式 3.3 に示す．*DV* の各要素は，要素の分子 *HitCount*(*k*, *k*+9)が HitCount(1,10)と比べてどの程度変化をしているのか示す値（変動率）となる．

$$DV = \left\{ \frac{HitCount(1,10)}{HitCount(1,10)}, \frac{HitCount(11,20)}{HitCount(1,10)}, \ldots, \frac{HitCount(991,1000)}{HitCount(1,10)} \right\} \ldots \text{数式 3.3}$$

短時間に，繰り返し「次へ」ボタンをクリックした場合に発生する変動に対する検証は，Yahoo! と Bing においてのみ行う．なぜなら，Yahoo!と Bing の検索 API においては検索結果をランキング上位 1,000 件まで取得できる一方，Google の検索 API では検索結果をランキング上位 64 件までしか取得することができないためである．

DV を各検索エンジン，各クエリにおいて取得した後，検索エンジンの変動をより明確にするために *DV* に対して k-means クラスタリングを行う．クラスタの数は 1 から 6 まで手動で変化をさせたのち，もっともよいクラスタリング結果を人手で選択する．また，クラスタリングを利用する際に利用する距離はコサイン類似度を用いる．

クラスタリングの結果を用いて，最終的に短時間に，繰り返し「次へ」ボタンをクリックした場合に発生する変動に対して考察を行う．

3.4 時間経過により発生する変動

本実験では，検索する日時を変えた場合のヒット数の変動に対する特徴をとらえるため，2009 年 10 月 11 日から，12 月 12 日までの約２ヵ月間，10,000 個のクエリを利用してヒット数の収集を行った．

収集したヒット数がどのように変動をしているのか調査をするために，*Vectors of Variational ratio* (*VV*) を定義する．*VV* の定義を数式 3.4 に示す．VV のそれぞれの要素は，10/11 に取得したヒット数に対する変化率を表す．

$$VV = \left\{ \frac{HitCount(10/11)}{HitCount(10/11)}, \frac{HitCount(10/12)}{HitCount(10/11)}, \cdots, \frac{HitCount(12/12)}{HitCount(10/11)} \right\} \quad \text{... 数式 3.4}$$

HitCount(*Date*)：*Date* に取得したヒット数

VV を各検索エンジン，各クエリにおいて取得した後，ヒット数の変動を明らかにするために k-means クラスタリングを行う．3.3 においてクラスタリングを行う際と同様に，クラスタの数を 1 から 6 まで手動で変化をさせたのち，もっともよいクラスタリング結果を人手で選択する．クラスタリングに用いる距離も，3.3 と同様にコサイン類似度を用いる．

クラスタリングの結果を用いて，最終的に時間経過により発生する変動に対して考察を行う．

4. 検証結果

本章では，3 章で述べた検証方法に従って検証を行った結果について述べる．短時間に，繰り返し同じクエリを利用して検索し場合に発生する変動に対する検証結果は 4.1 に，短時間に，繰り返し「次へ」ボタンをクリックした場合に発生する変動に対する検証結果は 4.2 に，時間経過により発生する変動に対する検証結果は 4.3 に記す．

4.1 短時間に，同じクエリを利用して検索した場合に発生する変動

3.2 に示した検証方法に従い，*CV* を求めた結果を表 4.1 に示す．表 4.1 においてクエリの合計数が 10,000 となっていないのは，検索時においてエラーが発生したクエリを取り除いているためである．表 4.1 より，Google は常にほぼ同一のヒット数を返すことが読み取れる．Google において *CV* の値が 0.0%にならなかったのは 9986 回のうち 9 回のみであり，その 9 回も *CV* の範囲が 0.0% < *CV* <= 0.1%と小さな範囲の中に収まっている．Yahoo!では，99.4%のクエリにおいて *CV* の範囲が 0.1%以下となっている．さらに，最大でも *CV* が 5%以下に収まっている．Bing では，Google，Yahoo!と比べるとヒット数の変動が激しい．しかしながら，97.5%のクエリにおいて *CV* は 0.5%以下であり，また *CV* が 20%を超えるクエリも 9,877 個中 1 個のみである．加えて，*CV* が 20%を超えたクエリについて確認をしたところ，影響を受けたクエリはアダルトワードであった．よって，この変動結果はセーフフィルタの影響を受けたのではないかと推測される．

この結果より，「短時間に，繰り返し同じクエリを利用して検索し場合に発生する変動」の影響は非常に少ないことがわかった．なぜならば，「短時間に，繰り返し同じクエリを利用して検索し場合に発生する変動」が発生をするとしても，多くの場合において *CV* が 5%以下となっているためである．

表 4.1 短時間に，繰り返し同じクエリを利用して検索し場合に発生する変動の分布

範囲	頻度		
	Google	Bing	Yahoo!
CV = 0.0%	9977	699	9096
0.0% < *CV* <= 0.1%	9	2555	730
0.1% < *CV* <= 0.5%	0	6191	46
0.5% < *CV* <= 1%	0	171	4
1% < *CV* <= 5%	0	56	1
5% < *CV* <= 10%	0	12	0
10% < *CV* <= 20%	0	4	0
20% < *CV* <= 100%	0	1	0
100% < *CV*	0	0	0
クエリの合計数	9986	9689	9877

4.2 検索開始オフセットの変化に伴う変動

3.3 に示した検証方法に従って，検索開始オフセットが変化した場合，つまり，短時間に繰り返し「次へ」ボタンをクリックした場合に発生する変動に対して検証を行う．

まず，*DV* を各検索エンジン，各クエリにおいて取得した後，検索エンジンの変動をより明確にするために *DV* に対して k-means クラスタリングを行った．クラスタのサイズを 1 から 6 に変動をさせクラスタリングを行ったのち，次の２つを判定基準としてもっとも良いクラスタサイズを人手で選択した．

・ヒット数の変動が開始する検索開始インデックスが異なっているか

・変動が起こった際，その変動率が増加したか，減少したか

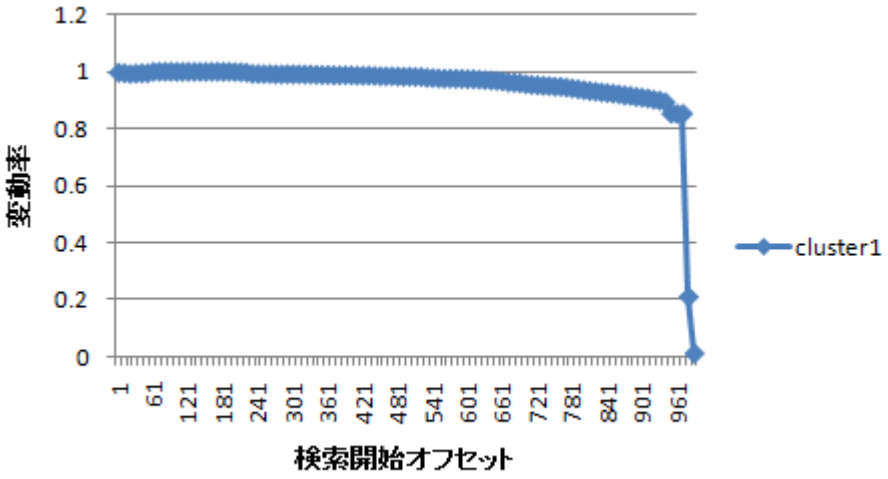


図 2. 繰り返し「次へ」ボタンをクリックしたときのヒット数の変動結果 (Bing)

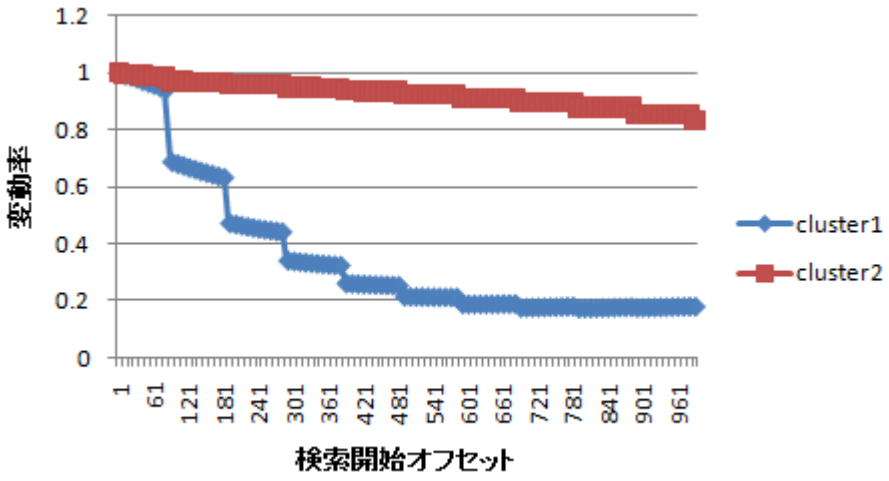


図 3. 繰り返し「次へ」ボタンをクリックしたときのヒット数の変動結果 (Yahoo!)

その結果，Bing において最もよいクラスタサイズは 1，Yahoo!において最もよいクラスタサイズは 2 となった．それぞれの結果のクラスタ平均値を図 2，図 3 に示す．また，クラスタリングを行った全ての結果を付録のⅡ，Ⅲに示す．

Bing における解析結果

Bing において唯一のクラスタである cluster 1 は，要素数が 3,152 個であった．クラスタ数が 1 であることからもわかるように, Bing においてはすべての *DV* が同様の傾向を示した．その傾向とは，検索開始オフセットがある程度大きな値，900 付近，になるまで一定，つまり *HitCount*(1,10)と等しくなり，その後急激にヒット数が実際に取得することのできた検索結果数まで減少する傾向である．これは，Bing が最終的な検索結果数を実際に取得可能な検索結果数に調整をしているため起こると考えられる．

Yahoo!における解析結果

Yahoo!の結果に表れた cluster 1 は cluster 2 よりも多い 7462 クエリを含んでいる．cluster 1 は検索開始オフセットが増加するのに伴って，徐々に減少する傾向をもっている．cluster 2 は 1,204 クエリを含んでおり，cluster 1 よりも早く減少する．変動率が約 0.2 となった検索オフセットで減少は止まり，その後ほぼ横ばいの安定したヒット数となる．この結果より，Yahoo!は *HitCount*(1,10)をまず算出した後，検索開始オフセットが増やされるにつれてヒット数を再計算していると考えられる．

ヒット数について考察をする際に，現在検索エンジンで利用されている高速化技術を考慮しなくてはならない．その高速化技術とは 2.1 で述べた分散処理，early termination，index pruning などである．これらの技術は検索の高速化に対して非常に大きな効果がある．その一方で，ヒット数の正確さに関しては高速化の副作用が表れることが想定される．例えば高速化技術を利用した場合, 検索結果の上位を取得する際にはクエリに一致する全ての Web ページを取得していないと考えられる．そのため，ヒット数は検索結果を表示するために収集した Web ページから推定し算出しているものと考えられる．ヒット数が検索結果を表示するために収集した Web ページより算出されているのならば，その収集した Web ページがより大きいほど，ヒット数の推定結果は正確になる．そのため，ヒット数は検索開始オフセットが大きな値であればあるほど，より正確なヒット数だと考えられる．ただし，ヒット数が実際に取得することのできた検索結果数に明らかに調整されている場合，その結果は意図的に調整されたものであるので信頼できるヒット数とは言えないと考えられる．

4.3　時間経過により発生する変動

3.4 に示した検証手法に従って，時間経過により発生する変動について検証した結果について述べる．

まず，*VV* を各検索エンジン，各クエリにおいて取得した後，検索エンジンの変動をより明確にするために *VV* に対して k-means クラスタリングを行った．クラスタのサイズを 1 から 6 に変動をさせクラスタリングを行ったのち，次の２つの基準を利用して人手によってもっとも良いクラスタサイズを選択した．

・ヒット数の変動が開始す検索の実行日が異なっているか

・変動が起こった際，その変動率が増加したか，減少したか

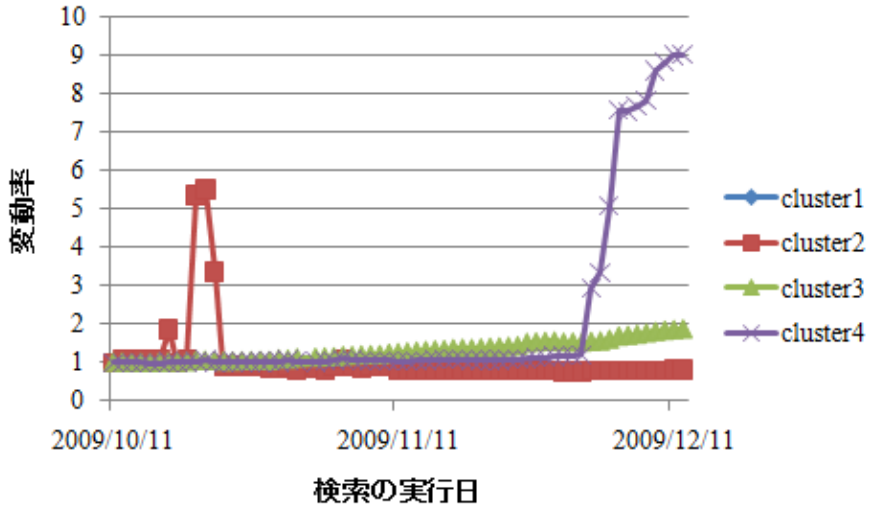


図 4. 検索日時の変化によるヒット数の変動（Google）

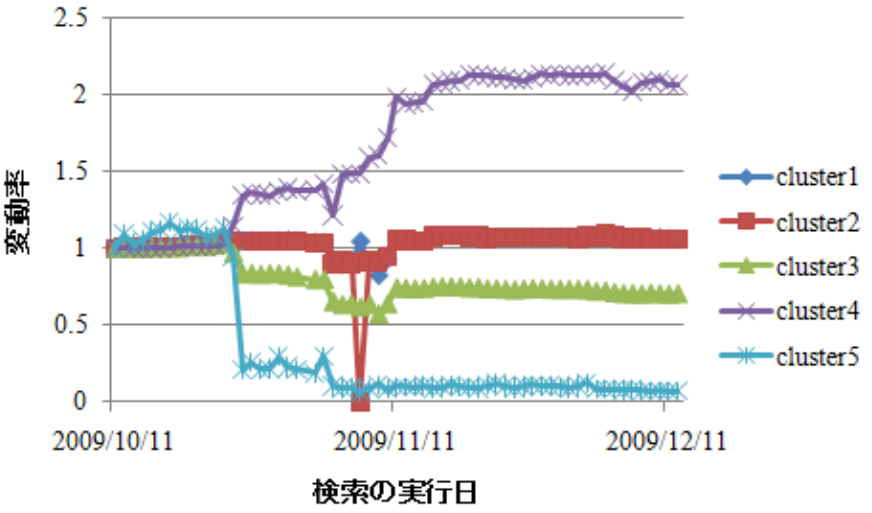


図 5. 検索日時の変化によるヒット数の変動（Bing）

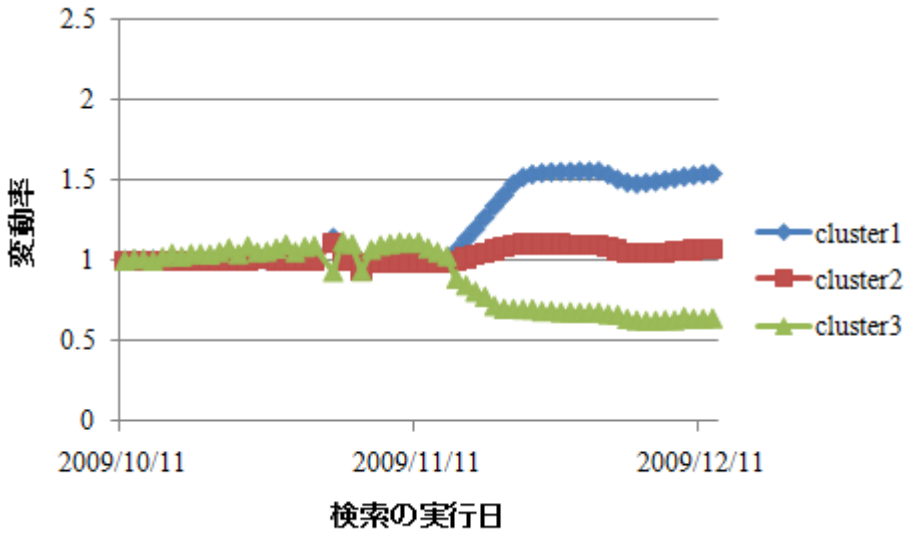


図 6. 検索日時の変化によるヒット数の変動（Yahoo!）

表 4.2 表各クラスタに含まれるクエリ数

	cluster1	cluster2	cluster3	cluster4	cluster5	合計
Google	7,167	979	644	133		8,923
Bing	4,998	1,640	1,279	478	67	8,462
Yahoo	4,587	2,020	217			6,824

その結果，Google において最もよいクラスタサイズは 4，Bing において最もよいクラスタサイズは 5，Yahoo!において最もよいクラスタサイズは 3 となった．それぞれの結果のクラスタ平均値を図 4，図 5，図 6 に示す．また，クラスタリングを行った全ての結果を付録の図Ⅴ～Ⅶに示す．

加えて，各クラスタの大きさをまとめた表を表 4.2 に示す．なお，人手によるクラスタリングの良さを判定する基準として，「ヒット数の変動が起こる検索実行日の違い」と「ヒット数の増減の傾向」を利用している．

Google における解析結果

Google において，cluster1 が最も大きなクラスタとなる．cluster1 は，10/11 のヒット数と比べて大きな変化がなく，ほぼ 1 の変動率を保っている． 2 番目に大きなクラスタである cluster2 は，10/20 から 10/22 の間に 1 度ヒット数が急上昇した後，再び変動率が 1 に戻っている．次に大きなクラスタである clulster3 は，日が経つにつれてヒット数が徐々に増加をしている．最後に，もっとも小さなクラスタである cluster4 は，12/2 を境にヒット数が急上昇をし，10/11 に対して約 8 倍に増加している．

Bing における解析結果

Bing では，全てのクラスタにおいてヒット数が 10/25 を境に大きく変化をしている．これは，Bing の検索インデックスが 10/25 に更新されたためではないかと推定される．その後，11/3～11/11 にかけてもヒット数が激しく変化をし，最終的に 11/11 以降ヒット数の変動が落ち着いている．この結果より Bing は，ヒット数の「変動期」と「安定期」を繰り返しているように見える．

Yahoo!における解析結果

Yahoo!では，11/13 から 11/20 にかけて変動が起こっていることを除いて，変動率がほぼ安定をしている．11/3 から 1 週間の間ヒット数が変動した理由として，検索インデックスの更新が考えられる．

3 検索エンジンの解析結果を踏まえた考察

上記の結果より，時間経過によるヒット数の変動は「変動期」と「安定期」の２通りに分類できる．変動期におけるヒット数は日々変化をしており，どの時点のヒット数が信頼できるヒット数なのか明らかではない．一方，安定期のヒット数は，ほぼ同じヒット数を返し続けるため，信頼できると考えられる．つまり，信頼できるヒット数を得るためには安定期のヒット数を得られれば良い．

ここで，どのような場合が変動期であり，逆にどのような場合が安定期かを知るために，変動期に注目をして考察を行う．Google，Bing の変動期に注目をすると，これらの変動は 1 日から４日以内に 30%以上の大きな変動が起きていることが読み取れる．これより，Google，Bing では，４日以内に 30%以上の変動が発生しなければ安定期であると考えることができる．また，Yahoo!での変動期に注目をすると，Google や Bing とは異なり，１週間をかけて徐々に 40%の変動が発生していることが読み取れる．これより Yahoo!では，１週間以内に 40%以上の変動が発生をしなければ，安定期であると考えられる．これを踏まえると，「1 週間の間」，「30%以上の変動が発生しなければ」安定期であると考えられる．

5. 信頼のできるヒット数

本章では，４章の実験結果を踏まえて，信頼できるヒット数の利用方法について提案を行う．

まず信頼性のあるヒット数を利用する際には，何度も「検索」ボタンを押すことによるヒット数の変動は考慮を行わなくてもよい．なぜなら 4.1 の結果より，何度も検索ボタンを押すことによるヒット数の変化の影響は小さいと考えられるためである．

その一方，繰り返し「次へ」ボタンを押すことによって発生するヒット数の変動と，時間経過によって発生するヒット数の変動を考慮しなくてはならない．4.2 の結果より，繰り返し「次へ」ボタンを押すことによって発生する Hit Count Dance の中からより信頼できるヒット数を求めるためには，検索開始オフセット k が最大の時の $HitCount(k,\ k+1)$を利用すればよい．ただし，検索エンジンによっては k が一定以上になるとヒット数が実際に取得可能な検索結果数に調整されてしまう．そのため，調整が行われた場合には調整が行われる直前のヒット数が最も信頼できるヒット数となる．

最後に 4.3 の結果より，時間経過によるヒット数の変動を避け，より信頼できるヒット数を求めるためには，ヒット数の変動が「安定期」のときのヒット数を利用すればよい．ヒット数が 1 週間以上，観測開始日に対して 30%以上増減しなければヒット数は安定期に入っている．

信頼できるヒット数についてまとめた表を表 5.1 に示す．

表 5.1 信頼できるヒット数

ヒット数の変動する状況	影響	信頼のできるヒット数が得られる場合
短時間に，繰り返し同じクエリを利用して検索する場合に発生する変動	なし	検索フィルタの影響を受けない場合
短時間に，検索開始オフセットを変更して検索をする場合に発生する変動	あり	「検索開始オフセットが最も大きい場合に得られるヒット数」でかつ「検索エンジンによってヒット数の調整が行われていない場合」
時間経過により発生する変動	あり	ヒット数が，１週間以上，観測開始日に対して 30%以上増減していない場合

6. おわりに

本研究では，ヒット数を研究に用いる基盤として，信頼性のあるヒット数の取得方法について検証を行った．

現在ヒット数は，自然言語処理分野をはじめとして様々な研究で利用されている．その一方で，ヒット数の特徴や信頼性については明らかにされていないため，ヒット数が研究に利用できるのかについて疑問視をされてきた．そこでヒット数の変動特徴や信頼性を解析するためにいくつか研究が行われてきた．既存の研究ではヒット数の特徴を明らかにし，変動がどのように発生しているのかについて検証を行っている．しかしながら，それらの検証は「実験が小規模である」，「時系列によるヒット数の変動を考慮していない」，「根拠が脆弱である」などが原因で十分な検証とはいえない．そこで本研究では，ヒット数が変動をする 3 つの状況として「短時間に，繰り返し同じクエリを利用して検索する場合」「短時間に，検索開始オフセットを変更して検索をする場合」「時間経過により発生する変動」を定義し，それぞれの状況がヒット数の変動にどのような影響を与えているのかについて検証を行った．そしてヒット数変動に対して検証を行った後に，信頼できるヒット数が得られる状況はどのような場合なのかについて提案を行った．

その結果として，信頼できるヒット数は「検索開始オフセットが最も大きい場合に得られるヒット数で，かつ，検索エンジンによってヒット数の調整が行われていない場合」かつ「1 週間以上，観測開始日に対して 30%以上増減していない場合」に得られることを確認した．

今後はより多くのクエリによる実験と，より長期間にわたるヒット数の観測を行うことで，提案した「信頼性のあるヒット数」が確かなものかどうか，より長期的な検証を行う必要がある．

関連研究

[1] A. Kilgarriff and G. Gefenstette. "Introduction to the Special Issue on the Web as Corpus", J. of Computational Linguistics, Vol.29, No.3, pp.333-347. (2003).

[2] R. L. Cilibrasi and P. M. B. Vitanyi. "The Google Similarity Distance", IEEE Trans. on Knowledge and Data Engineering, Vol.19, No.3, pp.370 - 383. (2007)

[3] Y. Matsuo, T. Sakai, K. Uchiyama and M. Ishizuka. "Graph-based Word Clustering using Web Search Engine", Proc. of the Conf. on Empirical Methods in Natural Language Processing, pp.542-550 (2006).

[4] A. Kilgarriff. "Googleology is Bad Science", J. of Computational Linguistics. Vol.33, No.1, pp.147-151 (2007).

[5] M. Thelwall. "Quantitative Comparisons of Search Engine Results", J. of the American Society for Information Science and Technology, Vol.59, No.11, pp.1702-1710 (2008).

[6] A. Uyar. "Investigation of the Accuracy of Search Engine Hit Counts", J. of Information Science, Vol.35, No.4, pp.469-480 (2009).

[7] L. Page, S. Brin, R. Motwani and T. Winograd, "The pagerank citation ranking: Bringing order to the web", Technical report, Stanford Digital Library Technologies Project(1998).

[8] A. Arasu, J. Cho, H. Garcia-Molina, A. Paepcke and S. Raghavan, "Searching the Web", ACM Trans. on Internet Technology, Vol.1, No.1, pp.2-43(2001).

[9] C. D. Manning, P. Raghavan and H. Schutze. "Introduction to Information Retrieval", Cambridge University Press, New York, NY.

[10] V. N. Anh, O. Krester and A. Moffat. "Vector-Space Ranking with Effective Early Termination", Proc. of the 24th Ann. Int'l ACM SIGIR Conf, pp.35-42 (2001).

[11] M. B. J. Jansen, A. Spink, T. Saracevic, "Real Live, Real Users, and Real Needs: A Study and Analysis of User Queries on the Web", Information Processing and Management, Vol. 36, No.2, pp.202-227(2000).

[12] D. Carmel, D. Cohen, R. Fagin, E. Farchi, M. Herscovici, Y. S. Maarek and A. Soffer. "Static Index Prunning for Information Retrieval Systems", Proc. of the 24th Ann. Int'l ACM SIGIR Conf. 43-50, 2001.

[13] 情報爆発時代に向けた新しい IT 基盤技術の研究. http://www.infoplosion.nii.ac.jp/info-plosion/

[14] Nutch，http://lucene.apache.org/nutch/

[15] Lucene，http://lucene.apache.org/java/docs/

謝辞

本大学で研究を行うにあたり，熱心にご指導・ご鞭撻を頂きました山名早人教授に深く感謝の意を表します. また, 研究生活のサポートをしてくださいました OB の平手勇宇先輩, ならびに黒木さやかさんを始めとする研究室の皆様，秘書の伊藤美代子さんに深く御礼申し上げます.

また本研究は，文部省科学研究助成費奨励研究 21300038（2009 年度）の補助の下行われた研究です．ここに記して謝意を表します.

業績一覧

査読あり論文誌

・舟橋卓也, 上田高徳, 平手勇宇, 山名早人: "商用検索エンジンの検索結果では取得できないランキング下位部分の収集・解析", 日本データベース学会論文誌, Vol.7, No.1, pp.37-42 (2008.6)

・舟橋卓也, 上田高徳, 平手勇宇, 山名早人: "商用検索エンジンのヒット数に対する信頼性の検証", 日本データベース学会論文誌, Vol.7, No.3, pp.31-36 (2008.12)

査読あり国際会議

・Takuya Funahashi, and Hayato Yamana. Hit Count Dance - Reliability Verification of Search Engines' Hit Counts, CIRSE2010 (2010.3)　（投稿中）

査読なし国内研究会

・舟橋卓也、上田高徳、平手勇宇、山名早人："商用検索エンジンのヒット数に対する信頼性の検証",情処研報(DBS)/iDB2008, Vol.2008, No.88, pp.139-144 (2008.9)

・舟橋卓也："iPod Touch, iPhone を利用したプレゼンテーション補助ツールの開発", 2008 年度先端 IT スペシャリスト育成プログラム 秋季研究プロジェクト 研究発表 (2009.1)

・舟橋卓也, 平手勇宇, 山名早人: "検索ヒット数のクラスタリングを用いた補正手法の提案", DEIM2009, i1-36 (2009.3)

・舟橋卓也, 曽根広哲, 山名早人："複数キーワードクエリに対する検索ヒット数の信頼性検証", 電子情報通信学会技術研究報告, Vol.109, No.153, pp.19-24（2009.7）

・舟橋卓也, 山名早人: "Hit Count Dance -検索エンジンのヒット数に対する信頼性検証-", DEIM2010 (2010.3 発表予定)

・山﨑 邦弘, 中村 智浩, 舟橋 卓也, 山名 早人：”Resizable-LSH：可変領域型の近似的類似検索”, 情報研報, Vol.2009-DBS-148 No.22 (2009.7.28)

受賞

・情報処理学会 データベースシステム研究会 学生奨励賞　(2008.9)

・情報処理学会 コンピュータサイエンス領域奨励賞 (2009.9)

付録

I. . 検証に利用したクエリのサンプル

mixi	マシェリ	自民党	kids
youtube	ebank	802	hi-ho
年賀状	音泉	人事院	マカロン
楽天	東京マルイ	サントノーレ	京都観光
you tube	ゲゲゲの鬼太郎	八方尾根	アントワン式
2	jal カード	ffftp	ハローワークインターネットサービス
2 ちゃんねる	明治	クライシスコア	ntn
郵便番号	下着	年賀素材	ジャパンレンタカー
google	和光堂	スギ薬局	賃貸情報
ニコニコ動画	魔法の i ランド	fujitsu	美輪明宏
goo	ガンバ大阪	銀行コード	duga
2ch	-	優木まおみ	お財布
ユーチューブ	鈴木えみ	http://www.aeoncredit.co.jp	9205
jal	xbox360	小林製薬	ラベンハム
ニコニコ	新築マンション	ピアノ	jeanasis
msn	フレッツ	ポインセチア	nicovideo
amazon	広島	hotjam	池袋西武
ana	東京スター銀行	sh905i	ダウニー
アマゾン	車検	porno tube	csi
グーグル	レイトン教授と悪魔の箱 攻略	ジョインベスト証券	セントフォース
jr	mk2	松田聖子	ピル
脳内メーカー	くら寿司	真鍋かをり	大きく振りかぶって
au	浅草	国会図書館	ozoc
郵便局	2008	ひぐらしのなく頃に解	サロモン
ハンゲーム	マジコン	アンリシャルパンティエ	河村隆一
cpz	c 型肝炎	八景島	丸三証券
dmm	マフラー	投資	バック
gyao	火山	アラド	社会福祉士
じゃらん	ペーパークラフト	スニーカー	不動産投資
jra	monex	軽自動車	major

hotmail	鉄拳 6	シナモン	sophia
ぐるなび	脳内メーカ	audi	winamp
nhk	西野公論	スペースワールド	amzon
ユニクロ	コート	エコキュート	rf
えろつべ	abc クッキング	national	資産運用
docomo	ラクーア	窓の社	zero
アダルト	jsg	セントラル	マレーシア
ハローワーク	野沢温泉	モノノ怪	愛媛県
ドコモ	ドラベース	ハチ北	テレビ大阪
価格	ユニバーサル	ero	p-line
地図	イリュージョン	福島県	長谷川潤
フジテレビ	ケンタッキーフライドチキン	goo 地図	歌詞画像
99bb	jfa	カーメイト	dragon ash
ウィキペディア	手帳	イエティ	radiohead
価格.com	ロックマン	アントワン定数	バレエ
宝くじ	ディオール	チョコレート	西川史子
メディアプロダクション ロゴ	お取り寄せ	esp	たからくじ
ホットペッパー	伊丹空港	ポケモンカード	ストレイテナー
郵便	ハニカム	里田まい	東京ウォーカー
psp	アクアタイムズ	花田美恵子	中小企業診断士

II. ． 検索開始オフセットの変化に伴う変動に対する クラスタリング結果（Bing）

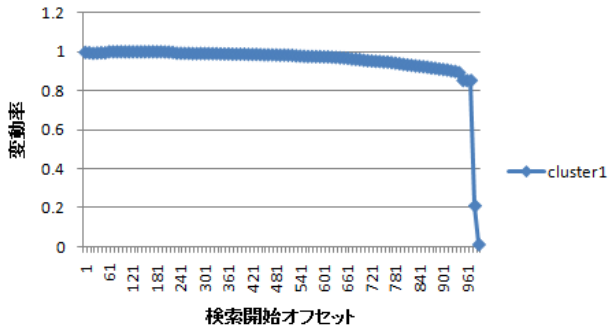


図 Ⅱ.1 クラスタサイズが 1 の時の結果

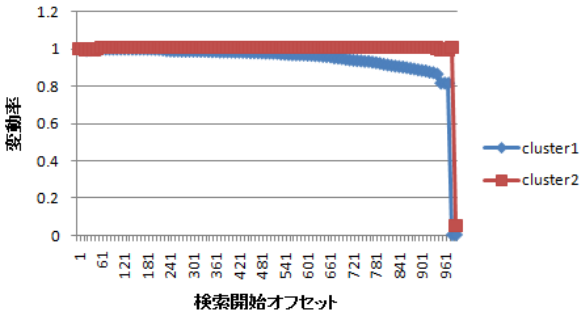


図 Ⅱ.2 クラスタサイズが 2 の時の結果

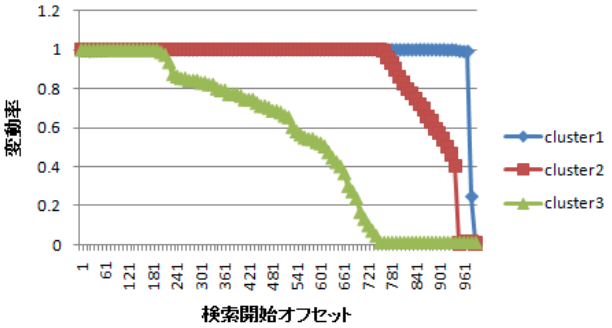


図 Ⅱ.3 クラスタサイズが 3 の時の結果

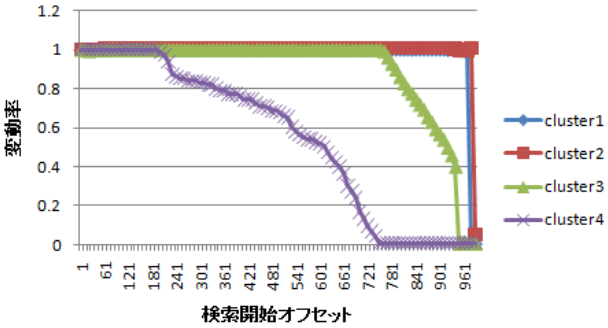


図 Ⅱ.4 クラスタサイズが 4 の時の結果

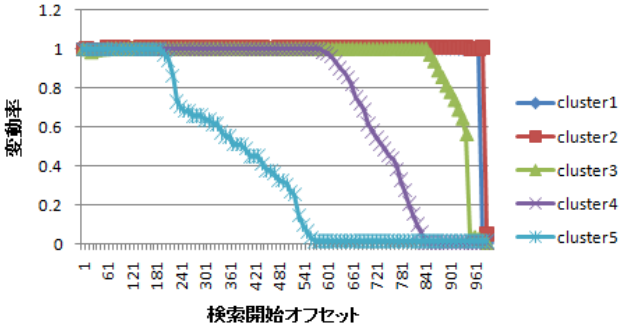


図 Ⅱ.5 クラスタサイズが 5 の時の結果

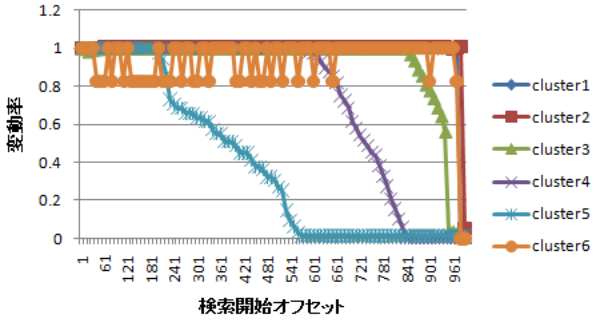
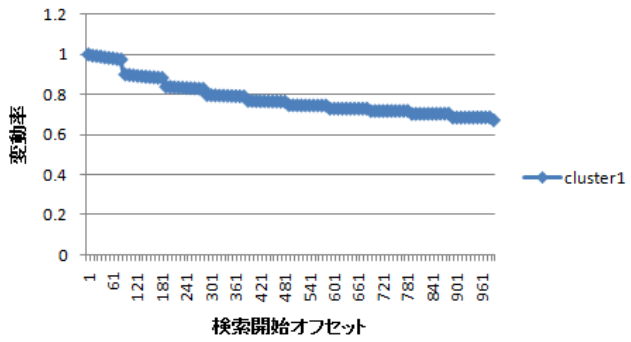
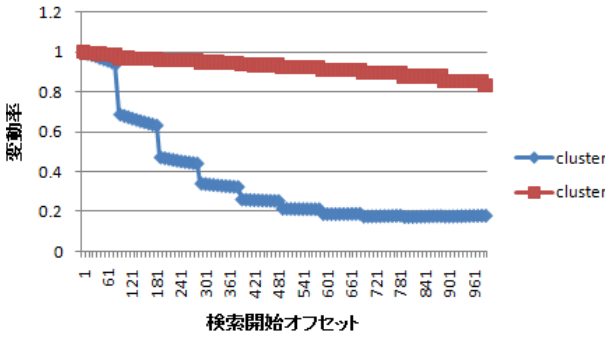


図 Ⅱ.6 クラスタサイズが 6 の時の結果

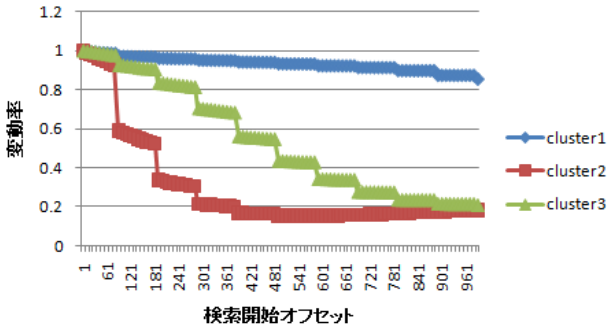
III. 検索開始オフセットの変化に伴う変動に対する クラスタリング結果（Yahoo!）



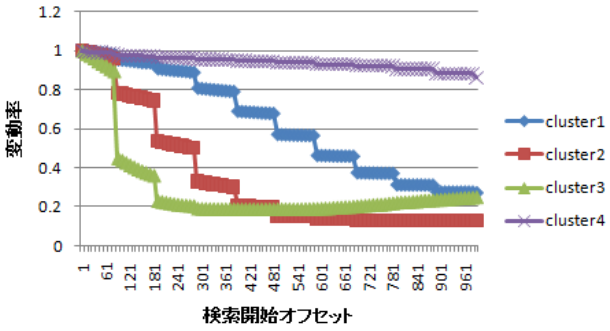
図Ⅲ.1 クラスタサイズが 1 のときの結果



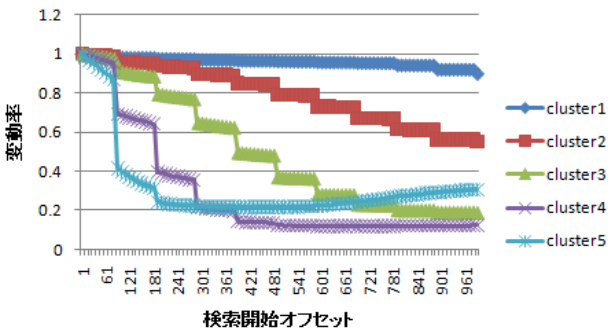
図Ⅲ.2 クラスタサイズが 2 のときの結果



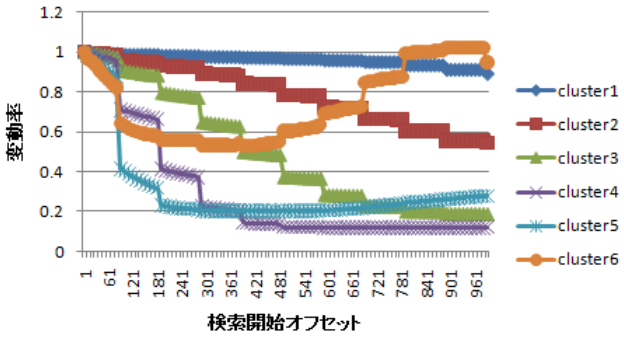
図Ⅲ.3 クラスタサイズが 3 のときの結果



図Ⅲ.4 クラスタサイズが 4 のときの結果



図Ⅲ.5 クラスタサイズが 5 のときの結果



図Ⅲ.6 クラスタサイズが 6 のときの結果

IV.. 付録Ⅰ，Ⅱにおける各クラスタリングの要素数

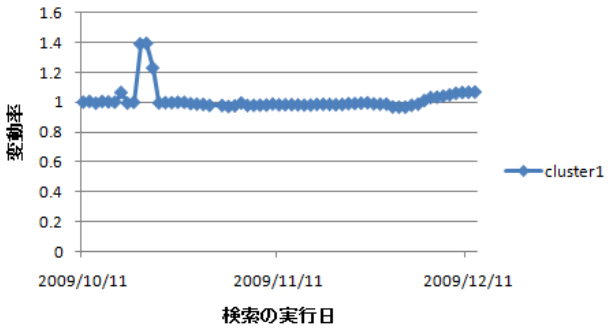
表Ⅳ．1 各クラスタの要素数（Bing）

	cluster1	cluster2	cluster3	cluster4	cluster5	cluster6	sum
k=1	3154						3154
k=2	2481	673					3154
k=3	2758	273	123				3154
k=4	2087	673	271	123			3154
k=5	2074	670	221	141	48		3154
k=6	2073	670	221	141	48	1	3154

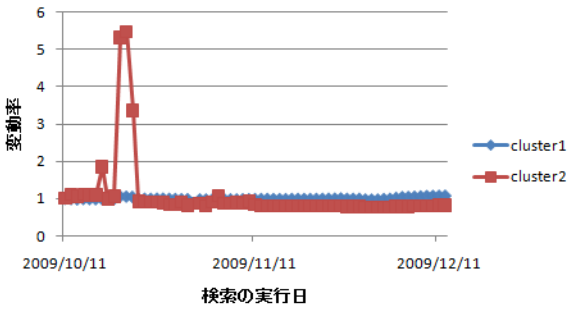
表Ⅳ．2 各クラスタの要素数（Yahoo!）

	cluster1	cluster2	cluster3	cluster4	cluster5	cluster6	sum
k=1	8671						8671
k=2	7467	1204					8671
k=3	7266	747	658				8671
k=4	7127	600	541	403			8671
k=5	6346	1026	518	476	305		8671
k=6	6271	978	513	456	305	148	8671

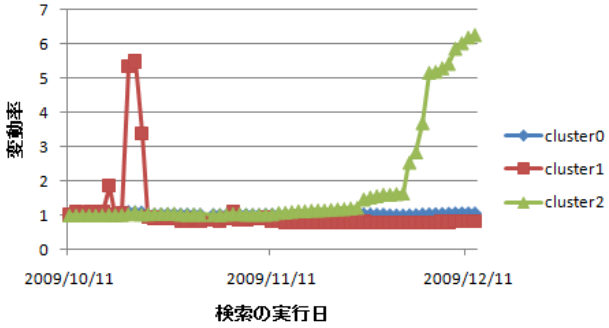
V. 時間経過により発生する変動のクラスタリング結果（Google）



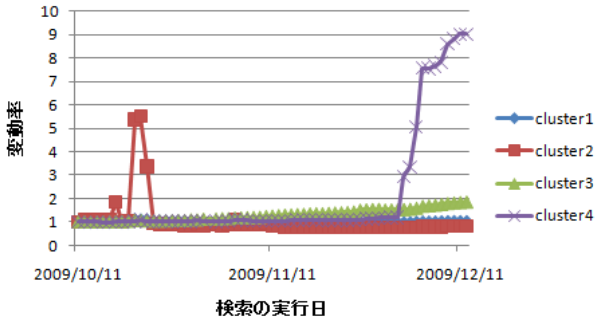
図V.1 クラスタサイズが 1 のときの結果



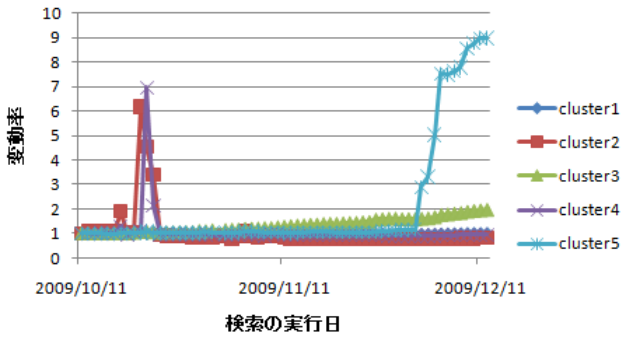
図V.2 クラスタサイズが 2 のときの結果



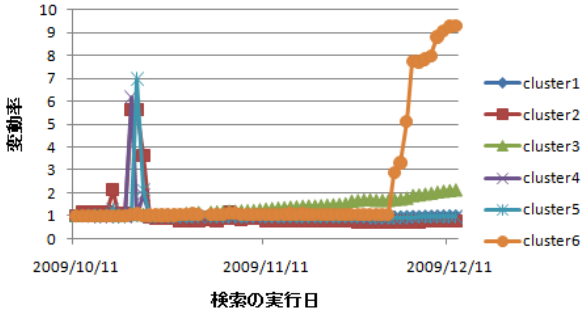
図V.3 クラスタサイズが 3 のときの結果



図V.4 クラスタサイズが 4 のときの結果

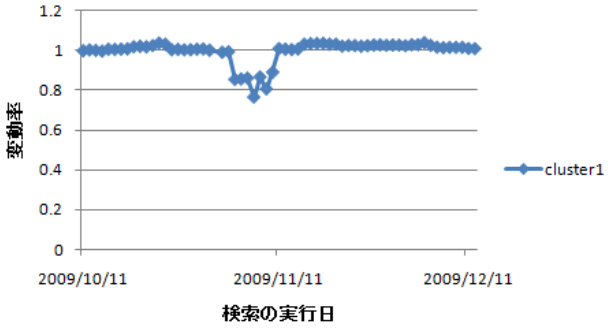


図V.5 クラスタサイズが 5 のときの結果

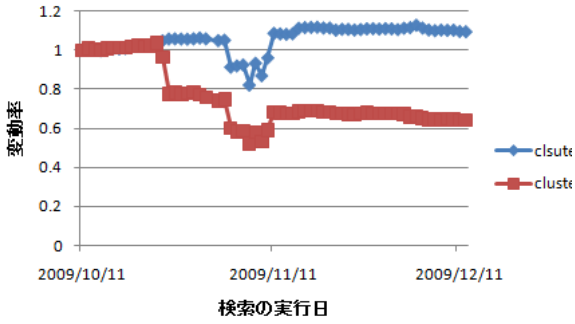


図V.6 クラスタサイズが 6 のときの結果

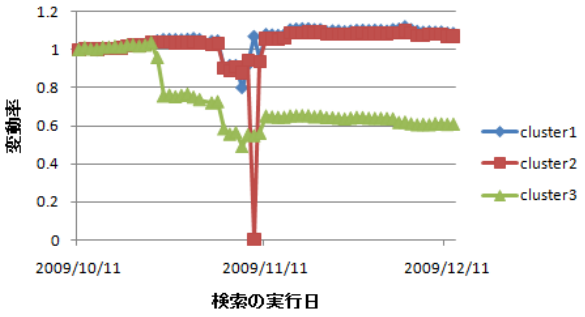
VI. 時間経過により発生する変動のクラスタリング結果（Bing）



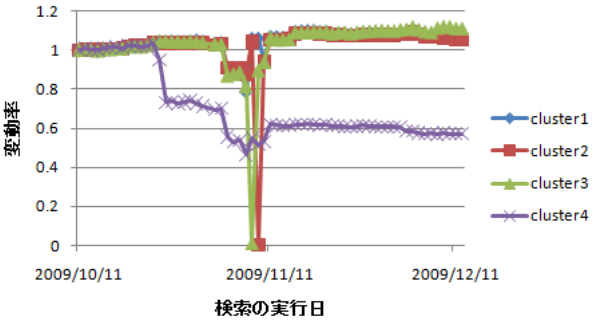
図VI.1 クラスタサイズが 1 のときの結果



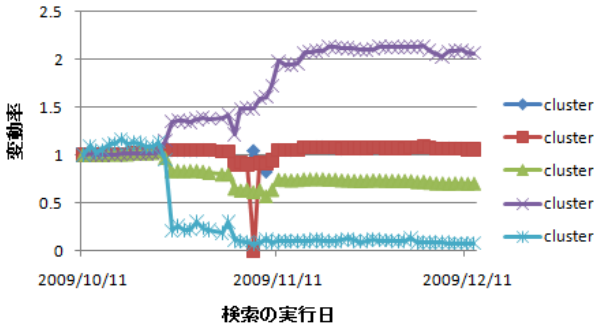
図VI.2 クラスタサイズが 2 のときの結果



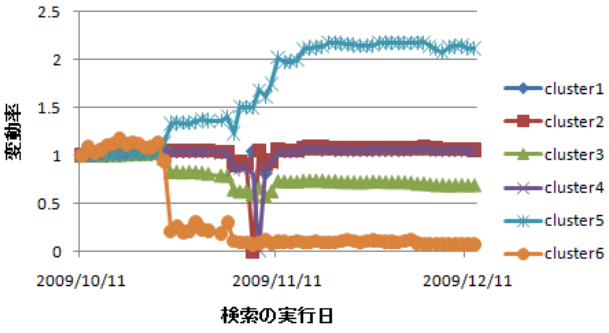
図VI.3 クラスタサイズが 3 のときの結果



図VI.4 クラスタサイズが 4 のときの結果

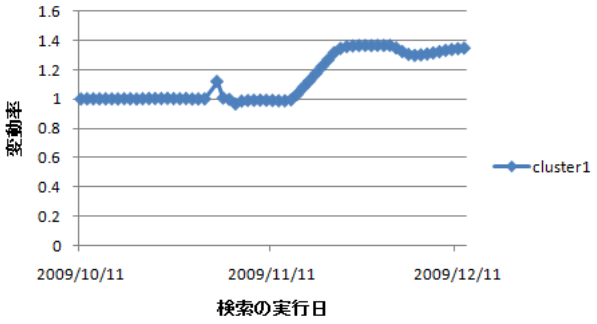


図VI.5 クラスタサイズが 5 のときの結果

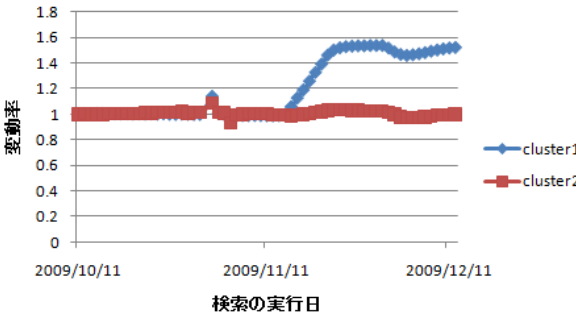


図VI.6 クラスタサイズが 6 のときの結果

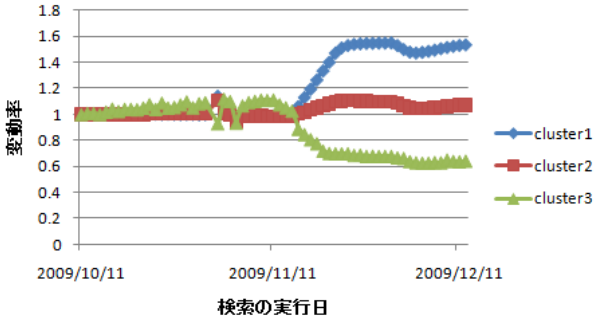
VII.　時間経過により発生する変動のクラスタリング結果（Yahoo!）



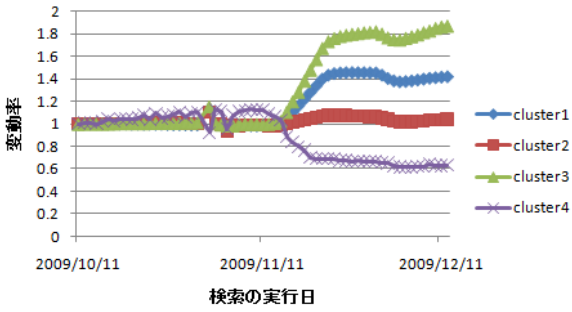
図Ⅶ.1 クラスタサイズが 1 のときの結果



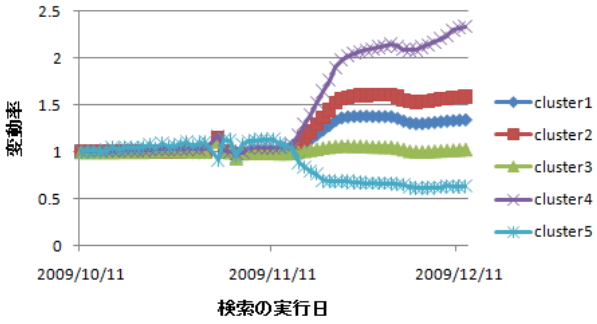
図Ⅶ.2 クラスタサイズが 2 のときの結果



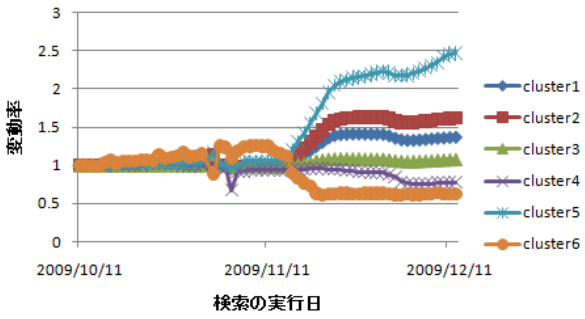
図Ⅶ.3 クラスタサイズが 3 のときの結果



図Ⅶ.4 クラスタサイズが 4 のときの結果



図Ⅶ.5 クラスタサイズが 5 のときの結果



図Ⅶ.6 クラスタサイズが 6 のときの結果

VIII.. 時間経過により発生する変動の各クラスタサイズ

表 Ⅶ.1 各クラスタの要素数（Google）

	cluster1	cluster2	cluster3	cluster4	cluster5	cluster6	Sum
k=1	8923						8923
k=2	7934	989					8923
k=3	7737	984	202				8923
k=4	7167	979	644	133			8923
k=5	7128	828	546	289	132		8923
k=6	7106	647	455	301	286	128	8923

表 Ⅶ.1 各クラスタの要素数（Bing）

	cluster1	cluster2	cluster3	cluster4	cluster5	cluster6	Sum
k=1	8462						8462
k=2	7233	1229					8462
k=3	5987	1493	982				8462
k=4	1355	1061	810	5236			8462
k=5	4998	1640	1279	478	67		8462
k=6	4395	1441	1157	962	441	66	8462

	cluster1	cluster2	cluster3	cluster4	cluster5	cluster6	Sum
k=1	6824						6824
k=2	4868	1956					6824
k=3	4587	2020	217				6824
k=4	3668	1739	1214	203			6824
k=5	2378	1596	2293	358	199		6824
k=6	2569	2062	1479	301	275	138	6824